

SPECIFICATION

TO ALL WHOM IT MAY CONCERN:

BE IT KNOWN THAT WE, KUNIHIO AKIYOSHI, a citizen of Japan residing at Fukuoka, Japan, HIROYUKI TANAKA a citizen of Japan residing at Fukuoka, Japan and MITSUO ANDO, a citizen of Japan residing at Fukuoka, Japan have invented certain new and useful improvements in

IMAGE FORMING APPARATUS AND METHODS USED
IN THE IMAGE FORMING APPARATUS

of which the following is a specification:-

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to an image forming apparatus that provides user services relating to image forming processes such as copying, printing, scanning, facsimile and the like.

2. Description of the Related Art

Recently, an image forming apparatus (to be referred to as a compound machine hereinafter) that includes functions of a printer, a copier, a facsimile, a scanner and the like in a cabinet is generally known. The compound machine includes a display part, a printing part and an image pickup part and the like in a cabinet. In the compound machine, three pieces of software corresponding to the printer, copier and facsimile respectively are provided, so that the compound machine functions as the printer, the copier, the scanner and the facsimile respectively by switching the software.

According to such a conventional compound machine, application programs corresponding to each of the capabilities such as printing, copying, faxing and scanning are launched. As for the existing application programs, since resource amounts to be used by the applications, and consistency of versions between

applications and the system of the compound machine and the like are known beforehand, the applications can be simply installed into the compound machine and can be launched. Therefore, for example, it is not necessary
5 to check resource amounts to be used by the applications before installing the applications or before launching the applications.

Since the conventional compound machine is provided with each software for the printer, the copier,
10 the scanner and the facsimile individually, much time is required for developing the software. Therefore, the applicant has developed an image forming apparatus (compound machine) including hardware resources, a plurality of applications, and a platform including
15 various control services provided between the applications and the hardware resources. The hardware resources include a display part, a printing part and an image pickup part. The applications perform processes intrinsic for user services of printer,
20 copier and facsimile and the like. The platform includes various control services performing management of hardware resources necessary for at least two applications commonly, performing execution control of the applications, and performing image forming
25 processes, when a user service is executed.

According to the compound machine, since the control services are provided separately from the applications, a new application that is developed by a user or a third party vendor can be installed in the compound machine. Therefore, there are following problems.

As for the existing applications such as the copy application, since the version and a necessary resource amount are known beforehand, there is no possibility of malfunction due to inconsistency of version or lack of resources when the application is installed and launched. However, as to the new application developed by the third vendor or the user, the status (version and available resources and the like) of the compound machine is not considered when developing the new application. Thus, there is a possibility that the operation of the compound machine becomes unstable when the new application is launched. More particularly, for example, when a necessary resource amount used by the new application is larger than the available resource amount in the compound machine, the operation of the compound machine may be stopped due to lack of the resource.

In addition, as for the existing applications for copying, printing, faxing and scanning, these

applications are configured such that they operate stably even when they are executed simultaneously in the compound machine. However, as for the new application developed by the third party, the new application is not necessarily developed in consideration of compatibility with other applications. Therefore, in a case when the new application and another application are mutually incompatible, if the new application is executed simultaneously with the another application, the new application operates abnormally or does not operate. For solving this problem, it can be considered to assign priorities to applications. However, the priorities should be updated each time when a new application is installed. In addition, for a general user, it is difficult to determine priorities of the applications.

In addition, among new applications developed by the third vendor, there may be a new application that should be executed with another application. If a user executes only the new application that needs the another application, the new application can not provide its full capabilities, or the application may be stopped.

25 SUMMARY OF THE INVENTION

A first object of the present invention is to provide an image forming apparatus and a method for obtaining application information for determining whether an application is installable. A second object of the present invention is to provide an image forming apparatus and a method for displaying the application information on an operation panel. A third object of the present invention is to provide an image forming apparatus and a method for performing launch restriction process according to information on resources. A fourth object of the present invention is to provide an image forming apparatus and a method for performing launch control for incompatible or indispensable applications.

The above-mentioned object is achieved by an image forming apparatus in which applications can be installed, in which the image forming apparatus includes: an application information obtaining part for obtaining application information that is used for determining whether an application is installable in the image forming apparatus.

According to the present invention, it can be determined whether an application is installable.

The above-mentioned object is also achieved by an image forming apparatus including an operation

panel for displaying or inputting operation information,
in which the image forming apparatus includes:

an information obtaining part for obtaining
application information on an application to be used in
5 the image forming apparatus; and

a display part for displaying the application
information or information related to the application
information on the operation panel.

According to the present invention, since
10 application information is displayed on the operation
panel, the user can determine whether an application is
installable.

The above-mentioned object is also achieved
by an image forming apparatus in which a plurality of
15 applications can be installed, in which the image
forming apparatus includes:

an information obtaining part for obtaining
necessary resource information for executing an
application and available resource information in the
20 image forming apparatus; and

a launch restriction part for comparing the
necessary resource information and the available
resource information, and performing launch restriction
process on the application according to the result of
25 comparison between the necessary resource information

and the available resource information.

According to the present invention, the image forming apparatus can prevent erroneous launch of an application for which resources are lacking. Thus, the
5 image forming apparatus can operate stably.

The above-mentioned object is also achieved by an image forming apparatus in which a plurality of applications can be installed, in which the image forming apparatus includes:

10 a part for referring to a database which includes, for each application, information on propriety of combination of applications; and

a launch control part for performing launch control process on an application according to the
15 information on propriety.

According to the present invention, proper applications can be launched according to the information on propriety of combination of applications.

20 BRIEF DESCRIPTION OF THE DRAWINGS

Other objects, features and advantages of the present invention will become more apparent from the following detailed description when read in conjunction with the accompanying drawings, in which:

25 Fig.1 is a block diagram of a compound

machine according to the first embodiment of the present invention;

Fig.2 is a block diagram of the hardware configuration of the compound machine according to the
5 first embodiment;

Fig.3 is a block diagram showing the configuration of the VAS 140, and relationship among the VAS 140, each application, the control service layer 150 and the general OS 121 according to the first
10 embodiment;

Fig.4 shows relationship between the VAS 140, and the hard disk 200 that stores application information, the application information table 106, and the NVRAM 104;

15 Fig.5 shows a whole procedure for performing install process or uninstall process in the compound machine 100;

Fig.6 is a flowchart showing a subroutine of application information obtaining process in step S401
20 in Fig.5;

Fig.7 shows an example of a program description (main function) of an application including the provisional launch function;

Fig.8 shows a sequence between the VAS and
25 the application;

Fig.9 shows an authentication screen of an initial setting screen according to the second embodiment;

Figs.10-16 show screens for file copy operations according to the second embodiment;

Figs.17-28 show screens for install operations according to the second embodiment;

Figs.29-36 shows screens for uninstall operations according to the second embodiment;

Fig.37 shows the configuration of the net file application according to the second embodiment;

Fig.38 is a block diagram showing the configuration of the VAS 140, and the relationship among the VAS 140, the applications, the control service layer 150 and the general OS 121 according to the third embodiment;

Fig.39 shows an example of the resource use information file 1201;

Fig.40 shows an example of the proc structure 211 that is referred to by the resource use information obtaining thread 1141;

Fig.41 shows a figure of a screen displayed on the operation panel in which launch restriction levels are shown by using a histogram;

Fig.42 shows a launch setting screen in which

launch setting has been performed for "simple scanner"
and "simple copy";

Fig.43 shows the launch setting screen in
which "simple print" is selected;

5 Fig.44 is a screen for advising to select
another application;

Fig.45 is an example of a screen in which
launch setting can be performed even if enough memory
is not included in the compound machine;

10 Fig.46 is a screen displayed when an
application for which any key is not assigned is
selected;

Fig.47 is a screen showing completion of
launch setting;

15 Fig.48 is a screen shown after the launch
setting ends;

Figs.49-53 shows screens in which release of
launch setting is performed;

Fig.54 shows a procedure performed by the VAS
20 140 when performing launch setting according to the
third embodiment;

Fig.55 is a flowchart for performing launch
restriction process when an application is launched
according to the third embodiment;

25 Fig.56 shows a flowchart for explaining the

operation of the VAS 140 when the resource use information file is produced;

Fig.57 is a block diagram of the compound machine of the fourth embodiment;

5 Fig.58 is a figure showing a configuration of the VAS 841-848 of the compound machine 800 of the fourth embodiment;

Fig.59 shows the configuration of the VAS 140 of the fifth embodiment, and shows relationship between
10 the VAS 140, each application, the control service layer 150 and the general OS 121;

Fig.60 shows an example of the application compatibility information file 2201;

Fig.61 shows another file according to the
15 fifth embodiment;

Fig.62 shows an example of a user selection screen for applications that can not be launched at the same time according to the fifth embodiment;

Fig.63 shows a flowchart of the procedure of
20 the application launch restriction process by the VAS 140 according to the fifth embodiment;

Fig.64 shows a flowchart of the procedure of the process of step S2505 according to the fifth embodiment;

25 Fig.65 shows a flowchart of the procedure of

launch restriction process according to the fifth embodiment;

Fig.66 is a figure showing a configuration of the VASes 941-948 of the compound machine 100 of the sixth embodiment, and relationship among the VASes 941-948, each application, control service layer 150 and general OS 121;

Fig.67 shows the configuration of the VAS 140 of seventh embodiment, and shows relationship between the VAS 140, each application, the control service layer 150 and the general OS 121;

Fig.68 is an example of the indispensable application information file 2202;

Fig.69 is an example of a screen for prompting the user to launch an indispensable application (plug-in) for a launch-requested-application (browser);

Fig.70 shows a flowchart of the procedure of the application launch control process by the VAS 140 according to the seventh embodiment;

Fig.71 shows a flowchart of the procedure of the process of step S2804 according to the seventh embodiment;

Fig.72 shows a flowchart of the procedure of launch control process according to the seventh

embodiment;

Fig.73 is a figure showing a configuration of the VASes 941-948 of the compound machine 100, and relationship among the VASes 941-948, each application, control service layer 150 and general OS 121;

Fig.74 shows a database of the eighth embodiment;

Fig.75 shows the procedure of launch control by the compound machine 100 according to the eighth embodiment;

Figs.76-81 show screens displayed when launch setting is performed according to the eighth embodiment.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

In the following, embodiments of the image forming apparatus and the method will be described with reference to figures.

(First embodiment)

Fig.1 is a block diagram of an image forming apparatus (to be referred to as a compound machine hereinafter) according to the first embodiment of the present invention. As shown in Fig.1, the compound machine 100 includes hardware resources and a software group 110. The hardware resources include a black and white line printer (B&W LP) 101, a color laser printer

102, and hardware resources 103 such as a scanner, a facsimile, a hard disk, memory (RAM, NV-RAM, ROM and the like) and a network interface. The software group 110 includes a platform 120, applications 130 and a
5 virtual application service 140 (to be referred to as VAS hereinafter).

The VAS 140 operates as a client process for the control services and operates as a server process for the applications. The VAS 140 obtains application
10 information before performing install process, in which the install process is performed for making an application launchable in the compound machine. Then, the VAS 140 produces an application information file in the hard disk 200. As an example of the information
15 that is obtained is product information. The product information includes a product ID, a vendor name, an application name, a version, a contact address (telephone number), required resources, determination on install, and related application information. The
20 product information can be included in an execution file of the application, or can be obtained by reading a file via a network for example. In addition, when the product information has been stored in the hard disk 200, a NVRAM 104, an application information table
25 106 or the like, the product information can be

obtained by reading one of them.

The application information included in the application execution file can be obtained by the VAS 140 by provisionally launching the application to
5 enable interprocess communication between the VAS 140 and the application.

By provisionally lunching the application, the application information in the execution file of the application can be obtained by the VAS 140 by using
10 interprocess communication between the VAS 140 and the application.

The application is copied to the hard disk from a PC card, IC card, SD card and the like, for example. Then, install process is performed for the
15 application, in which the application is set to be launchable. In the present embodiment, before performing the install process, the VAS obtains the application information that relates to the application, so that the VAS can determine whether the application
20 can be installed or not in the compound machine.

Whether the application can be installed or not is determined according to whether the application causes the compound machine to operate unstably or to stop its operation due to lack of resources or inconsistency of
25 versions or the like. The reason for performing the

check of the application before installing the application is that it is meaningless to install an application that cannot be launched, and that, the application may adversely affect the operation of the
5 compound machine if the application is erroneously launched..

The platform 120 includes control services for interpreting a process request from an application and issuing an acquiring request for hardware resources,
10 a system resource manager (SRM) 123 for managing one or more hardware resources and arbitrating the acquiring requests from the control services, and a general-purpose OS 121.

The control services include a plurality of
15 service modules, which are a system control service (SCS) 122, an engine control service (ECS) 124, a memory control service (MCS) 125, an operation panel control service (OCS) 126, a fax control service (FCS) 127, a network control service (NCS) 128. In addition,
20 the platform 120 has application program interfaces (API) that can receive process requests from the applications 130 by using predetermined functions.

The general purpose OS 121 is a general purpose operating system such as UNIX, and can execute
25 each piece of software of the platform 120 and the

applications 130 concurrently as a process.

The process of the SRM 123 is for performing control of the system and for performing management of resources with the SCS 122. The process of the SRM 123
5 performs arbitration and execution control for requests from the upper layer that uses hardware resources including engines such as the scanner part and the printer part, a memory, a HDD file, host I/Os (Centronics I/F, network I/F IEEE1394 I/F, RS232C I/F
10 and the like).

More specifically, the SRM 123 determines whether the requested hardware resource is available (whether it is not used by another request), and, when the requested hardware resource is available, notifies
15 the upper layer that the requested hardware resource is available. In addition, the SRM 123 performs scheduling for using hardware resources for the requests from the upper layer, and directly performs processes corresponding to the requests (for example,
20 paper transfer and image forming by a printer engine, allocating memory area, file generation and the like).

The process of the SCS 122 performs application management, control of the operation part, display of system screen, LED display, resource
25 management, and interrupt application control.

The process of the ECS 124 controls engines of hardware resources including the white and black laser printer (B&W LP) 101, the color laser printer (Color LP) 102, the scanner, and the facsimile and the like. The process of the MCS 125 obtains and releases an area of the image memory, uses the hard disk apparatus (HDD), and compresses and expands image data.

The process of the FCS 127 provides APIs for sending and receiving of facsimile from each application layer by using PSTN/ISDN network, registering/referring of various kinds of facsimile data managed by BKM (backup SRAM), facsimile reading, facsimile receiving and printing, and mixed sending and receiving.

The NCS 128 is a process for providing services commonly used for applications that need network I/O. The NCS 128 distributes data received from the network by a protocol to a corresponding application, and acts as mediation between the application and the network when sending data to the network. More specifically, the process of the NCS 128 includes server daemon such as ftpd, httpd, lpd, snmpd, telnetd, smtpd, and client function of the protocols.

The process of the OCS 126 controls an operation panel that is a means for transferring

information between the operator (user) and control parts of the machine. In the compound machine 100 of the embodiment, the OCS 126 includes an OCS process part and an OCS function library part. The OCS process
5 part obtains an key event, which indicates that the key is pushed, from the operation panel, and sends a key event function corresponding to the key event to the SCS 122. The OCS function library registers drawing functions and other functions for controlling the
10 operation panel, in which the drawing functions are used for outputting various images on the operation panel on the basis of a request from an application or from the control service. The OCS function library is dynamically linked to the application and each module
15 of the control services. All of the OCS 126 can be configured as a process, or can be configured as an OCS library.

The application 130 includes a printer application 111 that is an application for a printer
20 having page description language (PDL) and PCL and post script (PS), a copy application 112, a fax application 113, a scanner application 114 that is an application for a scanner, a network file application 115, a process check application 116 and the new applications
25 117 and 118. These applications are applications

specific for the compound machine 100. When the application is launched, the application sends an application registration request message with the process ID to the VAS 140. Then, the VAS 140 performs registration process for the application.

Interprocess communication is performed between a process of the application 130 and a process of the control service, in which a function is called, a returned value is sent, and a message is sent and received. By using the interprocess communication, user services for image forming processes such as copying, printing, scanning, and sending facsimile are realized.

As mentioned above, the compound machine 100 of the first embodiment includes a plurality of applications 130 and a plurality of control services, and each of those operates as a process. In each process, one or more threads are generated and the threads are executed in parallel. The control services provide common services to the applications 130. User services on image formation such as copying, printing, scanning and sending facsimile are provided while the processes are executed in parallel, the threads are executed in parallel, and interprocess communication is performed. A third party vendor can develop

applications for the compound machine 100, and can
executes the application in an application layer on the
control service layer in the compound machine 100. The
new applications 117 and 118 shown in Fig.1 are the
5 examples. The new application can be added or deleted
one by one.

Fig.2 shows an example of a hardware
configuration of the compound machine 100.

The compound machine 100 includes a
10 controller 160, an operation panel 175, a fax control
unit (FCU) 176, and an engine part 177 that is hardware
resource such as a printer that is specific for image
forming processing. The controller 160 includes CPU
161, a system memory 162, a north bridge (NB) 163, a
15 south bridge (SB) 164, ASIC 166, a local memory 167,
HDD 168, a network interface card (NIC) 169, a SD card
slot 170, a USB device 171, an IEEE1394 device 172, and
a Centronics 173. The memories 162, 167 may includes
RAMs and/or ROMs, for example. The FCU 176 and the
20 engine part 177 are connected to the ASIC 166 in the
controller via a PCI bus 178. The CPU 161 executes
programs of the application and control services and
the like installed in the compound machine 100 by
reading data from a RAM.

25 Fig.3 is a block diagram showing the

configuration of the VAS 140, and relationship among the VAS 140, each application, the control service layer 150 and the general OS 121.

The process of the VAS 140 includes a
5 dispatcher 145, a control thread 144 and an application information obtaining thread 141, and a screen display thread 142 and an application check thread 143. The screen display thread 142 produces an operation screen on the operation panel. The application check thread
10 143 checks whether the application can be installed according to the obtained application information.

The dispatcher 145 monitors message receiving from the application 130 or from control services. The dispatcher 145 sends a process request to the control
15 thread 144, the application information obtaining thread 141, the screen display thread 142 and the application check thread 143 according to the received message.

The control thread 144 requests the threads
20 141-143 to perform application information obtaining process, screen display process, and application check process. In the application check process, the VAS checks whether the application adversely affects the compound machine 100 if the application is launched.
25 For example, the VAS 140 obtains application

information of an application for which install is requested and obtains application information of applications that have already been installed. Then, the VAS can check resources or versions of the applications to check whether the application may adversely affect the compound machine. At that time, the screen display thread 142 displays application information of the application on the operation panel 105 by using application information stored in the application information file in the hard disk 200. Therefore, the user can grasp status of the application to be installed. Thus, even when an application is not determined to be installable, the user can understand the reason of the determination.

15 In addition, the control thread 143 controls install process and uninstall process, in which the install process is performed when the application is determined to be installable, and the uninstall process is performed when the application becomes unnecessary.

20 When the application information obtaining thread 141 receives a process request from the control thread 143, the application information obtaining thread 141 provisionally launches an application that is not installed so as to enable interprocess

25 communication between the VAS and the application, and

obtains application information from the application.
As to an application that is installed, by launching
the application, the application information can be
obtained by interprocess communication. The obtained
5 application information is stored in the hard disk 200
as the application information file 201. The
application information is recorded as a record for
each application.

The program of the VAS 140 can be provided as
10 a part or a whole of SDK (Software Development Kit).
The program is provided as an executable file or an
installable file in an recording medium such as a CD-
ROM or a FD. In addition, the program file of the VAS
140 can be provided via a network.

15 Fig.4 shows relationship between the VAS 140,
and the hard disk 200 that stores application
information, the application information table 106, and
the NVRAM 104. As shown in Fig.4, the product
information stored in the hard disk 200 is managed for
20 each application. By specifying a product ID,
application information corresponding to the product ID
can be specified. In addition, the hard disk 200
stores data of applications A and C that are copied in
the compound machine 100. The application B is
25 installed from an IC card. "F" that is indicated by

dotted lines indicates an execution file group existing in a path.

NVRAM 104 is a non-volatile memory that keeps memory content even when the power is turned off. The
5 NVRAM 104 stores the product ID, launch information, additional information for each application. When installing an application, information on the application is recorded in the NVRAM 104. In addition, the launch information indicates whether the
10 corresponding application is to be launched when the power of the compound machine is turned on.

The application information table stores information on applications as table data. By specifying the product ID of the application A, the
15 application name, the version and the install attribute can be obtained from the table, and the information can be displayed on the operation panel.

Fig.5 shows a whole procedure for performing install process or uninstall process in the compound
20 machine 100. Fig.6 is a flowchart showing a subroutine of application information obtaining process in step S401 in Fig.5.

In step S401 in Fig.5, the dispatcher 145 requests the application information obtaining thread
25 141 to obtain application information of an application

via the control thread 144. The application information obtaining thread 141 generates the application information file 201 by performing application information obtaining process shown in

5 Fig.6.

Then, the control thread 144 reads application information from the application information file 201, and produces an operation screen to display the operation screen on the operation panel
10 in step S402. The display screen displays the application name, the version, resource information and the like for each application. In addition, the display screen shows whether the application can be installed or not. By displaying the screen, the user
15 can understand status of applications, and the user can selects items in the screen to proceed install process.

After the user selects some items, the compound machine 100 waits for user input in step S403, and determines presence or absence of a user request in
20 step S404. The user request is an install request of an application or uninstall request of an application. If install of an application is requested and the request is permitted, information is registered to the NVRAM 104 shown in Fig.4 in step S405. If the request
25 is for uninstalling the application, application

information is deleted from the NVRAM 104 shown in Fig.4 in step S406.

Fig.6 is a flowchart showing the operation for obtaining the application information. Since the application information can not be obtained if the application is not launched, the VAS 140 provisionally launches the application in step S501. Then, the VAS 140 can obtain application information by requesting desired information from the application by using interprocess communication. According to the present embodiment, the VAS 140 obtains the product ID in step S502, and refers to the NVRAM 104, so that the VAS 140 can determine whether the application is already registered or not in step S503.

If the application is not registered in the NVRAM 104, the VAS 140 determines whether there is the product information file in step S504. If there is not the product information file, the VAS 140 obtains application information by using interprocess communication from the application in step S505. Then, the product information file is produced in step S506, and the VAS 140 determines whether the application can be installed or not in step S507. The VAS 140 reflects the determination result to the product information in step S508. For example, The determination is performed

by comparing resource information, obtained from the application, to be used by the application with available resource information of the system of the compound machine.

5 In step S509, on the basis of the determination result, it is determined that the application can be installed or not. When there is the product information file in the step S504, the application information is obtained from the product
10 information file in step S510. After information used for determining whether an application is installable is obtained in step S511, it is determined whether the application is installable in step S509.

 If the application is installable in step
15 S509, an installable flag is set in step S512, and the flag is written into the application information table in step S516. When the application is not installable, not-installable flag is set in step S513 and the flag is written into the application information table in
20 step S516. If the product information is already registered in the NVRAM 104 in step S503, the application information is obtained from the NVRAM 104 in step S514, and installed flag is set in step S515 and the flag is written into the application
25 information table in step S516.

If the above-mentioned process is not completed for every application, the step returns to the step S501, and the above-mentioned process is repeated.

5 According to the compound machine of this embodiment, since the application information is obtained for all applications, the status of applications can be grasped.

Next, the provisional launch of the
10 application will be described. The provisional launch is different from normal launch of the application in which substantive resources of the compound machine is used (launch for causing functions intended by the application is called "normal launch"). In the
15 provisional launch, the application does not obtain resources such as memory areas that are required when the application normally operates. In the provisional launch, the application only perform interprocess communication with the VAS 140, in which the
20 application provides, to the VAS, information on the application. The process of the provisionally launched application ends when communication between the application and the VAS ends. In addition, the function of the provisional launch of the application
25 is independent of normal functions of the application,

and is common to applications for the compound machines of the present embodiment. Therefore, for example, by providing a program template including the function of the provisional launch to a vendor, the vendor can
5 easily develop an application applicable for the compound machine by using the program template. For including application information in the application, for example, the vendor makes an include file in which the application information is included, and includes
10 the include file into the application when compiling the application program.

By providing the provisional launch function to the application, the application can provide application information to the VAS 140 without normal
15 launch, so that the VAS 140 can check the application.

Fig.7 shows an example of a program description (main function) of an application including the provisional launch function. This description can be provided as the program template. As shown in Fig.7,
20 whether the application is provisionally launched or normally launched is determined by an argument (-v). Accordingly, one of normal launch and provisional launch can be easily selected when the VAS 140 launches the application. That is, when provisional launch is
25 selected by using argument (-v), the application is

provisionally launched so that application information is notified. If provisional launch is not selected, the application is normally launched so that intrinsic operation of the application can be performed.

5 The VAS 140 obtains information from the application and determines whether the application is installable in this embodiment. Instead of this procedure, a procedure shown in Fig.8 can be adopted. In the procedure shown in Fig.8, after the VAS 140
10 determines that the application is installable, the VAS 140 asks the application whether the application is actually installable in step S601. If the application notifies the VAS 140 that the application is installable in step S602, the VAS 140 performs the
15 install process for the application in step S603.

Between the step S601 and the step S602, the application may obtains system information (resource data or configuration data) from the VAS 140, and the application may determines whether the application
20 itself is installable according to the system information.

In this embodiment, the VAS 140 obtains the application information from the execution file of the application. Instead of this method, for example, the
25 VAS 140 also can obtain the application information

from a file that includes the application information.
By storing the file in a server, the VAS 140 can obtain
the application information via a network.

In addition, displaying the screen on the
5 operation panel and checking applications can be
respectively performed by a program other than the VAS
140.

(Second embodiment)

In this embodiment, examples of screens
10 displayed by the screen display thread 142 can be
described in more detail.

Figs.9-36 show state transition of operation
screens displayed on the operation panel 105. Fig.9
shows an authentication screen of an initial setting
15 screen. Figs.10-16 show screens for file copy
operations. Figs.17-28 show screens for install
operations. Figs.29-36 shows screens for uninstall
operations.

In the authentication screen shown in Fig.9,
20 a user ID and a password are input. After input them
and the OK button 210 is pushed, a desired mode can be
selected. In this embodiment, an application is copied
in the compound machine 100 before installing the
application. As shown in Fig.10A, a file copy screen
25 is displayed, and a storing destination of a file to be

copied is selected. In this embodiment, a destination can be selected among from memory card 211, SD card 212 and via-network 213. As shown in Fig.10B, when via-network 213 is selected and the OK button 216 is pushed, a download screen shown in Fig.11A or a copy screen shown in Fig.11B is displayed. The download screen is for downloading an application by using Web. The copy screen is for copying an application by using LAN network. When cancel button 215 is pushed in Fig.10B, the screen is returned to Fig.10A. When the EXIT button 214 is pushed, the screen is returned to Fig.9 in which the authentication screen is not shown.

Fig.12A shows a case in which the memory card 211 is selected. Fig.12B shows a case in which the SD card 212 is selected. When the OK button is pushed, the screen is changed to Fig.13A in which a file list produce waiting screen is displayed. When the CANCEL button 221 is pushed, the screen is returned to Fig.10A. Fig.13B shows the file list 222.

Fig.14A shows a screen in which SimpleCopy 225 is selected from the file list. Fig.14B shows a screen in which SimplePrint 223 is selected additionally. When the OK button 216 is pushed in Fig.14A, the screen is changed to a waiting screen shown in Fig.15A. If there is the same file, a warning

screen shown in Fig.15B is displayed in which
confirmation for overwriting is displayed. When
copying of the file ends, a screen shown in Fig.16 is
displayed. At this time, copying of the application
5 into the compound machine is completed.

Fig.17 shows an example of an install
operation screen. When an install tag is selected, the
before-mentioned application information is obtained.

Fig.17A shows information such as
10 "registration allowed", "registration not allowed", and
"registered". The "registered" means that the
application information is recorded in the NVRAM by
performing install process. When the VAS 140 refers to
the NVRAM, the VAS 140 can determine whether the
15 application is already registered or not..

The "registration not allowed" is shown when
the VAS 140 determines that the application can not be
installed in the compound machine 100 for some reason.
For example, when a resource to be used by the
20 application is not provided on the compound machine 100,
the application is determined to be not-installable.

The "not registered (registration allowed)"
means that the install process is not yet performed
although the application is installable.

25 As shown in Fig.17B, when SimpleCopy 235 that

is an installable application is specified, the field of the application blinks. On the other hand, as shown in Fig.19A, when a registration-not-allowed application 234 is specified, the screen is changed to Fig.19B, and

5 information indicating the application cannot be installed and the reason are displayed. Information used for displaying this message was stored in the compound machine when the VAS 140 obtained application information.

10 In the screen shown in Fig.17A, when an application is selected for install, a medium that is an origination (a medium in which the application is copied) and a medium that is a destination in which the application exists after it is installed may be

15 displayed as shown in Fig.18. In this case, the destination can be selected. For example, as shown in Fig.18, an application copied in the SD card can be installed in the SD card.

When an installed application is selected,

20 the screen changes from Fig.20A to Fig.20B, and a warning screen is displayed. When an installable application is selected, after a registration waiting screen 240 shown in Fig.21A is displayed, a registration end screen shown in Fig.21B is displayed.

25 Then, application registration (application install)

ends. As a result, as shown in Fig.22, the number of installed applications increases.

Fig.23A shows another example in which installed applications and not-installable applications are shown. From the state of Fig.23B, when nothing is selected, the screen is returned to Fig,23A. As shown in Fig.24A, a plurality of applications can be selected as applications to be installed. As shown in Fig.24B, even if an application is selected, the selection can be canceled.

As shown in Fig.25A, when a not-installable application is selected, a warning screen is displayed as shown in Fig.25B. As shown in Fig.26A, when an installed application is selected, a warning screen shown in Fig.26B is shown, in which a warning indicating that setting values would be initialized is shown. As shown in Fig.27A, while the application is being registered, the screen is changed to a waiting screen. Then, an completion result is shown in Fig.27B. Then, as shown in Fig.28, application names that are installed are shown.

As to uninstall process, the screen shown in Fig.29A is shown. Fig.29B shows a screen in which an application to be uninstalled is selected. When an application for which uninstall is not allowed is

selected, warning screens shown in Figs.30A-30B are shown. When uninstall is allowed, Figs.31A and 31B are shown, and uninstall result is shown as shown in Fig.32.

Figs.33 and 34 are examples of uninstall
5 screens when a key is already assigned to the application. As shown in Fig.34A, when SimpleScan is selected, a screen shown in Fig.34B is shown in which whether key assignment is released and the application is uninstalled is inquired. When OK is pushed, the
10 screens shown in Figs.35A and 35B are shown, and uninstall result is shown in Fig.36.

As mentioned above, according to the compound machine 100 of this embodiment, copy process, install process and uninstall process can be easily operated by
15 using the operation screens shown in Figs.9-36. Since various pieces of application information are shown on the operation panel, the user can select screens sequentially according to displayed application information. As a result, easy operation can be
20 realized.

In the examples shown in Figs.9-36, the compound machine 100 determines whether an application is installable, and the determination result is shown on the operation panel. However, the compound machine
25 100 can be also configured such that the compound

machine 100 simply displays information necessary for determining whether an application is installable so that the user performs the determination. In this case, for example, version information of APIs used in the application and version information of APIs actually used in the compound machine may be displayed. In addition, a resource amount to be used by the application and an available resource amount that actually can be used in the compound machine may be displayed such that they can be compared.

The above-mentioned series of screens can be displayed on an apparatus such as a PC connected to the compound machine via network, and install process and the like can be performed from the apparatus.

In the case, the net file application 115 shown in Fig.1 has a Web server capability. A Java program that is a part of the net file application 115 generates a series of the screens and performs install process and the like for the compound machine 100 by using the so-called "servlet" method. The PC include a Web browser.

The configuration of the net file application 115 is shown in Fig.37. As shown in Fig.37, the net file application 115 includes a Web server 1151, a JVM (Java virtual machine) 1152 and a Java program 1153.

The net file application 115 receives a request from the PC. Then, the net file application 115 performs the install process and generates a HTML file and returns the HTML file to the PC.

5 The screen can be displayed on the PC simultaneously with displaying the same screen on the operation panel of the compound machine, or the screen can be displayed only on the PC. When the screen is displayed on the PC simultaneously with displaying the
10 same screen on the operation panel, for example, the net file application 115 obtains drawing information from the OCS, and converts the drawing information into HTML and sends the HTML.

(Third embodiment)

15 In the following, the third embodiment of the present invention will be described. The functional configuration of the compound machine 100 of the third embodiment is the same as that of the first embodiment. That is, the functional configuration of the compound
20 machine 100 of the third embodiment is shown in Fig.1.

As to a general personal computer, if a plurality of applications are launched one by one, there is a possibility that the PC hangs due to lack of resources. On the other hand, as to the compound
25 machine 100 of the present invention, the preinstalled

applications such as the copy application do not hang.

However, since the compound machine 100 can install a new application developed by a third vendor or a user, there may be a case in which lack of
5 resource occurs when launch setting is performed for a new application if a plurality of applications are already set to be launched.

The "launch setting" is performed for setting an installed application to be launched when the
10 compound machine 100 is powered on. That is, in the compound machine 100 of this embodiment, the application is not launched only by installing the application. It is necessary to perform launch setting on the application for launching the application. By
15 performing the launch setting, information indicating that the application is to be launched is stored in a memory. Then, when the compound machine 100 is powered on, the information is referred to so that the application is launched.

20 The VAS 140 obtains information on resources to be used by the application (also referred to as necessary resource information) from the application, and generates a resource use information file by using the information in the hard disk (HD 200). A text
25 memory area size, a heap area size and a stack area

size are examples of the information on resources to be used. The text memory area is a memory area in which programs of each application are loaded. The heap area is a memory area in which each application dynamically
5 keeps. The stack area is an area that is kept for storing arguments that are used for calling external modules by each application.

The information on the necessary resource amount is included in the execution file of the
10 application when developing the application. Then, the VAS 140 provisionally launches the application so as to obtain the information by using interprocess communication.

In addition, by storing the information on
15 resources to be used by the application in a file, the VAS 140 can obtain the necessary resource information by referring to the file.

Further, if the application has been actually executed on the compound machine 100, the VAS 140 can
20 obtain the necessary resource amount by obtaining the resource usage records of the application.

The VAS 140 also obtains an available resource amount in the compound machine 100 by subtracting resource amounts already used by
25 applications and system programs from the whole

resource amount of the compound machine 100. As the resource amounts already used by applications and system programs, usage records that are recorded in the compound machine can be used. The resource amounts
5 already used by applications and system programs can be recorded in the resource use information file, and the VAS 140 can refer to them.

At the time when the launch setting is performed for an application, the VAS 140 compares the
10 necessary resource amount for the application and the available resource amount. When the resource amount lacks in the compound machine, launch is restricted, that is, launch setting is not performed for the application. In addition to that, if the application
15 has a limited execution mode, the VAS 140 may set the application to run by the limited execution mode; in which the limited execution mode is a mode in which, for example, if the memory area is not enough, the application runs by using the hard disk area instead of
20 the memory area.

The resource check and the launch restriction can be performed not only when the launch setting is performed but also when the application is just actually launched.

25 Fig.38 is a block diagram showing the

configuration of the VAS 140 of this embodiment, and the relationship among the VAS 140, the applications, the control service layer 150 and the general OS 121.

In the process of the VAS 140, a dispatcher
5 144, a control thread 143, a resource use information obtaining thread 1141, and an application launch restriction thread 1142 operate.

The dispatcher 144 monitors message receiving from the application or the control services, and sends
10 a process request according to the message.

In a situation in which an application is actually launched, when the dispatcher 144 receives an application launch request message from the control service, the dispatcher 144 sends the received
15 application launch request message to the control thread 143. The control thread 143 sends a process request to the resource use information obtaining thread 1141 and the application launch restriction thread 1142.

20 In a situation in which launch setting is performed, for example, the dispatcher 144 sends a process request message to the control thread 143, and the control thread 143 sends a process request to the resource use information obtaining thread 1141 and the
25 application launch restriction thread 1142.

The control thread 143 can determine whether resource use information is recorded for an application that is installed by referring to the resource use information file 1201.

5 Then, the control thread 143 obtains the resource use information from the resource use information obtaining thread, and sends the resource use information to the application launch restriction thread 1142. The application launch restriction thread
10 1142 may refer to the resource use information file 1201. The application launch restriction thread 1142 compares the available resource amount and the necessary resource amount. If the necessary resource amount is larger than the available resource amount,
15 the application launch restriction thread 1142 performs launch restriction process.

 When the resource use information obtaining thread 1141 receives a process request from the control thread 143, the resource use information obtaining
20 thread 1141 refers to a proc structure (or u area 212) of the general OS 121 to obtain the text memory area size, heap area size and the stack area size used by an application. Then, the resource use information obtaining thread 1141 produces the resource use
25 information file 1201 in the hard disk (HD) 200. The

resource use information is recorded for each application as a record. Instead of this, as mentioned before, the resource use information obtaining thread 1141 may obtain necessary resource information from the application that is included in the execution file of the application, and record the resource information in the resource use information file 1201.

In addition, the resource use information obtaining thread 1141 can obtain resource amounts that are used by current processes from the proc structure 211, in which the currently used resource amounts are used for calculating the available resource amount.

Instead of performing the above-mentioned processing by using the threads, the VAS 140 may be configured as one process.

Fig.39 shows an example of the resource use information file 1201 stored in the HDD 200. As shown in Fig.39, the resource use information file 1201 stores the text memory area size, heap area size and the stack area size for each application ID (also referred to as product ID).

This file may store configuration data of the compound machine 100, for example, ADF. By doing so, launch restriction can be performed according to the configuration data. For example, if the compound

machine 100 does not have the ADF, launch of an application that uses the ADF may be restricted.

Fig.40 shows an example of the proc structure 211 that is referred to by the resource use information obtaining thread 1141. As shown in Fig.40, the proc structure 211 includes process ID (p_pid), text memory area size, heap memory area size and stack area size for each process. The proc structure is updated by the general OS when a process is executed, when a process ends, and when the status of the process changes.

In addition, the resource use information obtaining thread 1141 obtains resource amounts used by applications and system programs that are already set to be launched from the proc structure 211, and calculates the available resource amount by subtracting the currently used resource amounts from the whole resource amount of the compound machine.

When the application launch restriction thread 1142 receives a launch request for a new application or receives a launch setting request, the application launch restriction thread 1142 refers to the resource use information file 1201, and obtains information on resources to be used by the new application. In addition, the application launch restriction thread 1142 obtains the available resource

amount, for example, from the resource use information
obtaining thread 1141. Then, the application launch
restriction thread 1142 compares them and determines
whether launch of the new application needs to be
5 restricted.

In a case of launch setting, when the
application launch restriction thread 1142 determines
that application launch restriction is necessary, the
application launch restriction thread 1142 displays an
10 after-mentioned warning screen.

In a case of actual launching, when the
application launch restriction thread 1142 determines
that application launch restriction is necessary, the
application launch restriction thread 1142 sends a
15 launch restriction request message to an application
that sent the application launch request. When the
application launch restriction thread 1142 determines
that application launch restriction is not necessary,
the application launch restriction thread 1142 sends a
20 launch request message to the application so that
normal launch process is performed.

Fig.41 shows a figure of a screen displayed
on the operation panel in which launch restriction
levels are shown by using a histogram. This screen is
25 displayed when launch setting is performed. The

histogram shown in Fig.41 indicates the available resource amount that is obtained by subtracting resources used by applications and system programs that are preinstalled in the compound machine from the whole resource amount. That is, the figure visually shows how to divide remaining resources among new applications. By using the screen, the user can visually determine how to divide remaining resources among new applications. The histogram shows a resource amount to be used by a user application 221 that is set to be launched first, and resource amount to be used by a user application 222 that is set to be launched secondly. As for a user application 223 for which launch setting has not been performed, predicted resource amount is shown. In the example of Fig.41, if launch setting is performed for the user application 223, the resource amounts reach a dangerous level. Thus, launch setting can not be performed for the user application 223. In the screen, the launch restriction levels are shown by using colors, for example, safe (blue), warning (yellow) and danger (red). In addition, at the position of the dangerous level, the histogram may blink, or a voice guidance or warning sound may be output.

Examples of screens displayed on the

operation panel will be shown in the following. Each screen is displayed by the VAS 140, for example.

Fig.42 shows a launch setting screen in which launch setting has been performed for "simple scanner" and "simple copy". As shown in the figure, memory amounts to be used by the "simple scanner" and "simple copy" are shown in the histogram 251.

Next, when "simple.print" is selected for performing launch setting as shown in Fig.43, the part of "simple.print" blinks for a second. Then, the value of the histogram becomes 100% and the histogram blinks.

After that, screens shown in Figs.44 and 45 are displayed. Fig.44 is a screen for advising to select another application. By pushing the OK button, the screen returns to Fig.42. Fig.45 is an example of a screen in which launch setting can be performed even if enough memory is not included in the compound machine. When an application for which any key is not assigned is selected, a screen shown in Fig.46 is displayed.

In the screen of Fig.42, when "simple scanner" is selected and when launch setting ends, a screen shown in Fig.48 is displayed after the screen of Fig.47. As shown in Fig.48, the value of the histogram is increased by the memory amount to be used by the

"simple scanner".

When releasing launch setting for an application, screens shown in Figs.49-53 are displayed. In the state of Fig.49, when "simple scanner" is
5 selected for releasing launch setting, "simple scanner" blinks as shown in Fig.50. If OK is pushed, a screen shown in Fig.51 is displayed. Then, if OK is pushed, a screen of Fig.53 is shown via Fig.52. As shown in Fig.53, a memory amount of the simple scanner is
10 decreased from the histogram.

The application launch restriction process performed by the VAS 140 of the compound machine 100 will be described with reference to a flowchart of Fig.54. Fig.54 shows a procedure performed by the VAS
15 140 when performing launch setting.

When the VAS 140 receives a launch setting request for an application in step S1601, the VAS 140 checks whether there is resource use information for the application by referring to the HD 200 in step
20 S1602.

When the resource use information file exists, the VAS 140 obtains the resource use information from the resource use information file 1201 in step S1603. If the resource use information file does not exist in
25 step S1602, the VAS 140 produces the resource use.

information file in step S1604. For example, the control thread 143 requests the resource use information obtaining thread 1141 to produce the resource use information file, so that the resource use
5 information obtaining thread 1141 produces the resource use information file.

The procedure for producing the resource use information file will be described later with reference to Fig.56.

10 Next, the VAS 140 obtains the available resource amount in step S1605. For calculating the available resource amount, resource amounts of applications for which launch setting has already been performed can be obtained from the proc structure by
15 using a system call or a service function call. The system call is provided by the general OS, and the service function is provided by the control service layer 150. In addition, if the resource amounts are included in the resource use information file 1201, the
20 resource amounts can be also obtained from the resource use information file 1201.

Next, the VAS 140 compares the necessary resource amount to be used by the application for which launch setting is requested with the available resource
25 amount in step S1606. The application launch

restriction thread 1142 determines whether the necessary resource amount is larger than the available resource amount in step S1607. If the necessary resource amount is larger than the available resource amount, a screen shown in Fig.44 is displayed in step S1608. If the necessary resource amount is not larger than the available resource amount, launch setting is normally performed in step S1609.

If the application has a mode in which the application runs by using a hard disk area instead of a memory area, a screen for asking the user whether to use the mode can be displayed. When the user selects the mode, the application is executed by using the mode. The VAS 140 can know that the application has the mode by referring to the application information.

Fig.55 is a flowchart for performing launch restriction process when an application is launched.

When the dispatcher 144 receives an application launch request message from the control service layer, the dispatcher 144 send the message with a process ID to the control thread 1143. The control thread 143 determines whether the application that is requested to launch is an application other than the preinstalled applications such as the copy application, that is, whether the application that is requested to

launch is a new application in step S1701. If the application is the new application in step S1702, the control thread 143 checks whether there is resource use information for the application by referring to the HD
5 200 in step S1703.

When the resource use information file exists for the application, the control thread 143 obtains the resource use information from the resource use information file 1201, and sends the resource use
10 information to the application launch restriction thread 1142 in step S1704. If the resource use information file does not exist in step S1703, the control thread 143 requests the resource use information obtaining thread 1141 to produce the
15 resource use information file, so that the resource use information obtaining thread 1141 produces the resource use information file 1201 for the new application in step S1705. After the file is produced, the step S1704 is performed.

20 Next, the VAS 140 obtains the available resource amount in step S1706.

Next, the VAS 140 compares the necessary resource amount to be used by the new application for which launch is requested with the available resource
25 amount in step S1707. The application launch

restriction thread 1142 determines whether the necessary resource amount is larger than the available resource amount in step S1708. If the necessary resource amount is larger than the available resource amount, a screen of warning is displayed in step S1709. If the application has the mode in which the application runs by using a hard disk area instead of a memory area, a screen for asking the user whether to use the mode can be displayed. When the user selects the mode, the application may be executed by using the mode.

If the necessary resource amount is not larger than the available resource amount, the new application is normally launched in step S1710.

Fig.56 shows a flowchart for explaining the operation of the VAS 140 when the resource use information file is produced.

First, the VAS 140 provisionally launches applications from which the VAS 140 wants to obtain resource information in step S1801. Then, the VAS 140 obtains resource information from the application by using interprocess communication in step S1802. Then, the VAS 140 produces the resource use information file 1201 by using the obtained resource information in step S1803. Fig.39 is an example of the resource use

information file in which text memory area size, heap area size and stack area size are recorded for each application ID.

If the application has been launched and the
5 actual record of resource usage of the application is stored in the proc structure 211, the VAS 140 can obtain the usage record instead of obtaining information from the application. By updating the resource use information file by using the actual
10 records, updated resource information can always be obtained, that is, learning capability is provided. If the actual usage record changes each time when obtaining the record, the mean value or the maximum value of the records can be used as recourse
15 information to be recorded in the resource use information file 1201.

In the case in which the resource use information file 1201 is produced by using the proc structure 211 in the RAM 210, when the resource use
20 information obtaining thread 1141 receives the process ID, the product ID and the resource use information obtaining request message from the control thread 143, the resource use information obtaining thread 1141 refers to the proc structure 211, and searches for a
25 position of a block of the process ID of the

application ID. Then, the resource use information
obtaining thread 1141 obtains the resource use
information from the block of the process ID, and
records the obtained information in the resource use
5 information file 1201 with the application ID.

According to the compound machine of this
embodiment, it can be prevented that the compound
machine operates unstably due to lack of resources. In
addition, the new applications developed by the user or
10 the third vendor can be executed safely.

In this embodiment, the VAS 140 performs the
resource use information obtaining process and the
application launch restriction process for all
applications 130. However, the VAS 140 can be also
15 configured so as to cover a part of applications. For
example, the VAS 140 may perform the resource use
information obtaining process and the application
launch restriction process only for the new
applications 117 and 118.

20 In addition, considering that the resource
use amount of the new application may change, the VAS
140 may perform launch restriction on the new
application even when the necessary resource amount is
smaller than the available resource to some extent.

25 (Fourth embodiment)

The compound machine 100 of the third embodiment includes one VAS for all applications. According to the compound machine of the fourth embodiment, a plurality of VASes are launched for each application in which each VAS performs the resource use information obtaining process and the application launch restriction process for a corresponding application.

Fig.57 is a block diagram of the compound machine of the fourth embodiment. As shown in Fig.57, the compound machine 800 is different from the compound machine 100 of the third embodiment in that a plurality of virtual application services 841-848 operate for each application.

The VASes 841-848 performs the resource use information obtaining process and the application launch restriction process for a corresponding application for the printer application 111, copy application 112, fax application 113, the scanner application 114, the net file application 115 and the process check application 116 and the new applications 117 and 118.

Fig.58 is a figure showing a configuration of the VAS 841-848 of the compound machine 800 of the fourth embodiment, and relationship among the VAS 841-

848, each application, control service layer 150 and general OS 121.

According to the compound machine 800 of the fourth embodiment, different from the compound machine 100 of the third embodiment, as shown in Fig.58, a VAS control process (daemon) 801 operates between the VASes and each application. The VAS control process 801 generates the VASes 841-848 for each corresponding application. The operation of the VAS control process 801 is similar to that of the control thread 143.

Each process of the VASes 841-848 includes the dispatcher 145 and the threads 1141 and 1142. The function of each thread of the compound machine 800 is the same as that of corresponding thread of the third embodiment. According to the fourth embodiment, the processing for each application can be performed in parallel, so that the launch restriction process can be performed efficiently.

According to the third and fourth embodiments, the text memory area, the heap memory area, and the stack memory area are examples, and other resources can be used for application launch restriction.

(Fifth embodiment)

In the following, the fifth embodiment of the present invention will be described. The functional

configuration of the compound machine 100 of the fifth embodiment is the same as that of the first embodiment. That is, the functional configuration of the compound machine 100 of the third embodiment is shown in Fig.1.

5 In this embodiment, the VAS 140 obtains application compatibility information and produces an application compatibility information in an hard disk (after-mentioned HD 200), in which the application compatibility information indicates incompatible
10 applications for each application. As to mutually incompatible applications, exclusive control is required.

 The following example pairs of applications require exclusive control. An example is a pair of
15 applications of the same type such as a pair of authentication applications, and a pair of billing applications. In addition, each of a pair of printer applications and a pair of display applications require exclusive control, since the applications of each of
20 the pairs may scramble the same device (printer or display part). Further, a pair of the same applications of different versions requires exclusive control.

 In addition, exclusive control is necessary
25 for the following cases.

As a first case, exclusive control is required for an application that requires large memory area. The exclusive control is also required for an application that uses very large CPU power.

5 In addition, exclusive control is necessary for applications that generate RAW files of different formats in order to prevent problems due to format mismatch.

 In addition, as to an authentication or
10 billing application, there are applications suitable for a specific authentication or billing application. For example, in a case in which a scanner application A and a copy application A are suitable for an authentication application A, and a scanner application
15 B and a copy application B are suitable for an authentication application B, exclusive control is necessary between the authentication application A and the scanner/copy applications B if authentication methods of the authentication applications A and B are
20 different. In addition, exclusive control is necessary between the authentication application B and the scanner/copy applications A.

 In addition, exclusive control is necessary between a log obtaining application and a copy
25 application wherein the log obtaining application

produces thumbnails of copied images, and the copy application instantaneously deletes image data in the hard disk.

Further, for example, if there is an arrangement that applications of vendor A and vendor B can not be executed simultaneously in an compound machine, exclusive control is necessary.

When the VAS 140 obtains the application compatibility information, the VAS 140 also obtains attribute information of each application. The attribute information is used, for example, for displaying a screen.

In addition, the VAS 140 obtains the application compatibility information by provisionally launching the application by using interprocess communication. The VAS 140 may obtain the application compatibility information from an external server.

When launch setting is performed or when an application is launched, the VAS 140 detects incompatible applications according to the application compatibility information so as to perform launch control. For example, if incompatible applications are detected, one application is launched and another application is not launched.

Fig.59 shows the configuration of the VAS 140

of the fifth embodiment, and shows relationship between the VAS 140, each application, the control service layer 150 and the general OS 121.

The process of the VAS 140 includes a control
5 thread 143, an application compatibility information obtaining thread 2141 and an application launch restriction thread 2142. Processing by the VAS 140 can be also performed without using the threads.

The dispatcher 144 monitors message receiving
10 from an application or a control service. The dispatcher 144 sends a process request to the control thread 143, the application compatibility information obtaining thread 2141 and the application launch restriction thread 2142 according to the received
15 message. In this embodiment, when the dispatcher 144 receives application launch request message for applications, the dispatcher 144 sends the message to the control thread 143. The control thread 143 sends a process request to the thread 2141 and the thread 2142.

20 Application launch restriction process of this embodiment is execution control process in which only one application in a pair of applications is launched and another application is not launched.

When the application compatibility
25 information file 2201 does not include application

compatibility information for all installed applications, the control thread 143 can request the thread 2141 to obtain necessary application compatibility information. The obtained application
5 compatibility information is stored in the application compatibility information file 2201.

When the application compatibility information file 2201 includes necessary application compatibility information, the control thread 143
10 requests the application launch restriction thread 2142 to perform application launch restriction according to the application compatibility information.

The application compatibility information obtaining thread 2141 obtains application compatibility
15 information by using interprocess communication, and stores the information in the application compatibility information file 2201. In addition, if the application compatibility information can be obtained via a network, the application compatibility information obtaining
20 thread 2141 receives necessary information from the network by using the NCS 128. The application compatibility information is recorded as a record for each application.

The application compatibility information
25 file 2201 can be stored in the compound machine 100 by

the user. In addition, when the application compatibility information file 2201 can be downloaded from a server, the VAS 140 can obtain the application compatibility file 2201 from the server.

5 Although the above-mentioned operation of each threads is an operation when an application is launched, similar operation can be performed when launch setting is performed. Application launch restriction process for the launch setting is, for
10 example, displaying mutually incompatible applications and performing launch setting only for one selected application.

Fig.60 shows an example of the application compatibility information file 2201 stored in the HDD
15 200. As shown in the figure, the file includes, for each application ID (can be referred to as product ID), one or more application IDs that are incompatible with an application. In addition, the HD 200 stores a file shown in Fig.61. The file shown in Fig.61 includes,
20 for each application ID, application attribute, operation target, application name (file name), version and the like. The information can be used for displaying a screen, and also can be used for
25 determining which application should be launched among incompatible applications.

The application launch restriction thread 2142 refers to the application compatibility information file 2201, determines whether there are incompatible applications. If there are incompatible applications, the application launch restriction thread 2142 performs execution control in which only one application is launched among the incompatible applications. In this process, it is necessary to determine which application to launch when execution control is performed. For example, by assigning higher priority to an application having a newer version for applications having the same name, an application can be automatically determined to launch among incompatible applications by using information shown in Fig.61. Alternatively, by displaying the incompatible applications on the operation panel, the user can select an application.

Fig.62 shows an example of a user selection screen for applications that can not be launched at the same time. As shown in Fig.62, the screen shows a launch requested application list 2211, an incompatible application list 2212 and a radio button 2212 used when user selects an application. The user can select an application by using an operation key.

When the application launch restriction

thread 2142 determines that application launch restriction is necessary, the application launch restriction thread 2142 sends a launch restriction request message to an application that can not be
5 launched. On the other hand, when the application launch restriction thread 2142 determines that launch restriction is not necessary, the thread 2142 sends a launch request message to each application, so that the applications are normally launched. In addition, when
10 launch restriction is necessary, a launch request message is sent to an application to be launched according to the exclusive control.

Next, the application launch restriction process by the VAS 140 will be described. Fig.63 shows
15 a flowchart of the procedure.

When the dispatcher 144 receives an application launch request message from the control service layer 150 in step S2501, the dispatcher 144 sends the message to the control thread 143 with a
20 process ID of the application. The control thread 143 determines whether launch is requested for a plurality of applications in step S2502. When a plurality of applications are requested to launch, it is determined whether the application compatibility information file
25 2201 exists in step S2503. When the file exists, the

control thread 143 refers to the file in step S2504.
When the file does not exist, the application
compatibility information file is generated in step
S2505, and the generated file is referred to. The
5 process of the step S2505 will be described later with
reference to Fig.64.

The application launch restriction thread
2142 determines whether there are a pair of
incompatible applications among the launch requested
10 applications in step S2506. If such pair exists,
launch restriction process is performed in step S2507.
The launch restriction process will be described in
detail later with reference to Fig.65.

If there is no pair of incompatible
15 applications, normal launch process is performed in
step S2508.

Fig.64 is a flowchart for explaining the
process of generating the application compatibility
information file 2201.

20 First, the control thread 143 determines
whether application compatibility information can be
obtained via a network in step S2601. When the
information can not be obtained via the network, the
VAS 140 provisionally launches the application that is
25 requested to launch in step S2602. Then, the VAS 140

obtains the application compatibility information in step S2603 from the application. Then, the file 201 for the application is produced by using the application compatibility information in step S2604.

- 5 The application compatibility information for an application corresponds to a record of the file shown in Fig.60.

As mentioned before, instead of producing the application compatibility information file 2201, the
10 user may prepare the application compatibility information file 2201 and store it in the compound machine 100.

Fig.65 is a flowchart of the launch restriction process. This flowchart corresponds to the
15 step S2507 in Fig.63.

When launch restriction is necessary for a pair of incompatible applications, it is determined whether one of the pair of the applications is launched or launch is not performed for both of the applications
20 of the pair in step S2701. For example, this determination can be performed according to initial setting. When one application is launched (Yes in step S2701), the VAS 140 determines whether there is a selection criterion in step S2702. The selection
25 criterion may be predetermined priorities, and may be

recorded in the application compatibility file. In addition, the version can be used as an example of a criterion, in which an application of a newer version is to be selected when the names of the applications of the pair are the same.

When there is no selection criterion, the VAS 140 displays a selection screen including an incompatible application list on the operation panel in step S2703. Fig.62 shows an example of this screen.

10 The launch requested application list 2211 shows three application names with the versions. The incompatible application list 2212 shows two incompatible applications (verification AP and authentication AP) with other information (version, attribute, target),

15 and radio buttons are provided for selection in the left side. The user refers to information displayed on the operation panel so that the user can easily select an application to be launched by checking a radio button in step S2704.

20 As mentioned above, when the user selects an application in step S2704, execution control is performed in which launch of applications that are not selected is restricted in step S2705.

In the step S2702, if there is a selection

25 criterion for an application to launch, an application

selected according to the criterion is launched, and launch of other applications is restricted in step S2706.

5 If launch of all incompatible applications is restricted in step S2701, or if the user does not select any application in step S2704, all of the incompatible applications are not launched in step S2707.

10 Although the above-mentioned example relates to actual launch of applications, exclusive control can be also performed when launch setting is performed. In the case of launch setting, launch setting is performed instead of an application is actually launched.

15 As mentioned above, according to the compound machine 100 of the this embodiment, since exclusive control is performed to prevent incompatible applications from being launched simultaneously, the compound machine 100 operates stably.

20 In addition, according to the compound machine of the this embodiment, when the application compatibility information is obtained via a network, the application compatibility information file can be easily updated to include newest information. Thus, a new application can be supported.

25 (Sixth embodiment)

The compound machine 100 of the fifth embodiment includes one VAS for all applications. According to the compound machine of the sixth embodiment, a plurality of VASes 941-948 are launched
5 for each application in which each VAS performs the application compatibility information obtaining process and the application launch restriction process for a corresponding application.

Fig.66 is a figure showing a configuration of
10 the VASes 941-948 of the compound machine 100 of the sixth embodiment, and relationship among the VASes 941-948, each application, control service layer 150 and general OS 121.

According to the compound machine 100 of the
15 sixth embodiment, different from the compound machine 100 of the fifth embodiment, as shown in Fig.66, a VAS control process (daemon) 901 operates between the VASes and each application. The VAS control process 901 generates the VASes 941-948 for each corresponding
20 application. The operation of the VAS control process 901 is similar to that of the control thread 143.

Each process of the VASes 941-948 includes the dispatcher 145 and the threads 2141 and 2142. The function of each thread of the compound machine 100
25 of this embodiment is the same as that of corresponding

thread of the fifth embodiment. According to the sixth embodiment, the processing for each application can be performed in parallel, so that the launch restriction process can be performed efficiently.

5 (Seventh embodiment)

 In this embodiment, the compound machine 100 can simultaneously launch applications of indispensable relationship or recommended relationship. Applications of indispensable relationship means that executing one
10 application is indispensable for executing another application. Applications of recommended relationship means that executing one application is recommended for executing another application.

 The configuration of the compound machine 100
15 this embodiment is similar to that of the compound machine 100 of the fifth embodiment. The functions of the VAS 140 are different between the fifth embodiment and this embodiment.

 The VAS 140 of the seventh embodiment refers
20 to indispensable application information file 2202, and identify an application that can be executed solely, and an application that requires another application to run. The indispensable application information file 2202 may be produced by the VAS 140, or may be stored
25 in the compound machine 100 by the user. In addition,

the compound machine 100 can obtain the indispensable application information file 2202 from a server via a network.

For example, in a case when a new capability
5 can be provided by adding another application (plug-in software) to an application, indispensable application information in the indispensable application information file 2202 may be information on the plug-in software. However, the indispensable application
10 information is not limited to this. The indispensable application information can be any information on pair of applications that can be executed simultaneously for providing a capability.

The VAS 140 can obtain indispensable
15 application information via a network, and also can obtain the information from an application by provisionally launching the application.

The VAS 140 obtains indispensable application information corresponding to an application that is
20 requested to launch. If there is an indispensable application for the application, the VAS 140 controls the compound machine 100 to launch both of the application and the indispensable application. For example, if launch of the indispensable application is
25 not requested yet, launch of the application

corresponding to the indispensable application is not performed. Alternatively, for example, when the indispensable application is installed in the compound machine 100, the indispensable application is

5 automatically launched with the application. Further, for example, notification to urge the user to launch the indispensable application may be output. The notification is, for example, information display on the operation panel, voice guidance, warning sound and

10 the like.

Fig.67 shows the configuration of the VAS 140 of this embodiment, and shows relationship between the VAS 140, each application, the control service layer 150 and the general OS 121.

15 The process of the VAS 140 includes a dispatcher 144, a control thread 143, an indispensable application information obtaining thread 2145, and an application launch control thread 2146. The function of the dispatcher 144 is the same as that of the fifth

20 embodiment.

When the indispensable application information file 2202 does not include indispensable application information for an application, the control thread 143 can request the thread 2145 to obtain

25 necessary indispensable application information. The

obtained information is stored in the indispensable application information file 2202.

When the indispensable application information file 2202 includes necessary indispensable application information, the control thread 143 requests the application launch control thread 2146 to perform application launch restriction according to the indispensable application information read from the file 2202.

10 Like the fifth embodiment embodiment, the indispensable application information obtaining thread 2145 obtains indispensable application information by provisionally launching the application or via network, and generates the indispensable application information file 2202. Alternatively, the indispensable application information file 2202 can be prepared and stored in the compound machine 100 by the user. In addition, the compound machine 100 may obtain the indispensable application information file 2202 from a server.

Fig.68 is an example of the indispensable application information file 2202 stored in the HD 200. As shown in Fig.68, the indispensable application information file 2202 records, for each application ID, presence or absence of indispensable application,

application type, application ID of indispensable application.

When the application launch control thread 2146 receives an application launch request from the control service layer 150, the application launch control thread 2146 refers to the indispensable application information file 2202, and determines whether information on an application that is requested to launch is recorded in the file 2202. If the information is recorded, it is determined whether there is an indispensable application corresponding to the application. If the indispensable application does not exist, the application is normally launched. If the indispensable application exists, the VAS 140 performs launch control for the application and the indispensable application. In Fig.68, the browser (ID:103) and the plug-in (ID:105) are mutually indispensable.

When information on the application that is requested to launch is not recorded in the indispensable application information file 2202, the application launch control thread 142 requests the indispensable application information obtaining thread 2145 to obtain information on the application.

Fig.69 is an example of a screen for

prompting the user to launch an indispensable application (plug-in) for a launch-requested-application (browser).

Next, the application launch control process performed by the VAS 140 will be described. Fig.70 shows a flowchart of the procedure.

When the dispatcher 144 receives an application launch request message from the control service layer 150 in step S2801, the dispatcher 144 sends the message to the control thread 143 with a process ID of the application. The control thread 143 checks whether information on the launch requested application is recorded in the indispensable application information file 2202 in step S2802. When the information is recorded, the file 2202 is referred to in step S2803.

If the information is not recorded in the file 2202 in step S2802, the file 2202 is prepared in step S2804, and the file 2202 is referred to in step S2803. The process of the step S2804 will be described later with reference to Fig.71.

The application launch control thread 2146 determines whether there is an indispensable application corresponding to the launch requested application by referring to the indispensable

application information file 2202 in step S2805. If the indispensable application exists, launch control process is performed in step S2806. If the indispensable application does not exist, the launch requested application is normally launched in step S2807. The launch control process for the application and the indispensable application will be described in detail later with reference to Fig.72.

Fig.71 is a flowchart for explaining the process of generating the indispensable application information file 2202.

First, the control thread 143 determines whether the indispensable application information can be obtained via a network in step S2901. When the information can not be obtained via the network, the VAS 140 provisionally launches the launch requested application in step S2902. Then, the VAS 140 obtains necessary information in step S2903. Then, the file 2202 for the application is produced by using the obtained information in step S2904.

Fig.72 is a flowchart showing launch control process for a pair of the launch-requested-application and the corresponding indispensable application.

As shown in Fig.72, first, the application launch control thread 2146 determines whether launch of

the indispensable application is requested in step S3001. If Yes, both of the application and the indispensable application are normally launched in step S3008.

5 In step S3002, it is determined whether the launch of the launch-requested-application should be restricted until the indispensable application is requested to launch. By storing information, in the file 2202, indicating whether the application should be
10 launched or not until the indispensable application is launched, the VAS 140 can perform the determination of step S3002 by referring to the file 2202.

 If the launch of the launch requested application should be restricted, a notification is
15 output for prompting launch of the indispensable application in step S3003. As an example of the notification, an error message is displayed on the operation panel as shown in Fig.69. After outputting the notification, the process returns to the step S3001,
20 and the VAS 140 determines whether launch is requested for both of the application and the indispensable application. After the step S3003, if the user launches both of the application and the indispensable application, the applications are normally launched in
25 step S3008 (Yes in step S3001).

In step S3002, if it is determined that launch of the launch requested application is not restricted, it is determined whether the corresponding indispensable application has been installed in the compound machine 100 in step S3004. If the indispensable application has been installed, the indispensable application is automatically launched normally in steps S3007 and S3008.

If the indispensable application has not been installed in step S3004, it is determined whether the indispensable application is installable in step S3005. If installable, the indispensable application is installed and launched (steps S3006-S3008).

If the indispensable application is not installable in step S3005, the process ends since both of the applications can not be launched. At this time, the VAS 140 can display a screen indicating that the indispensable application is not installable.

Although the above-mentioned process is performed when the application is actually launched, the similar process can be performed when launch setting is performed for the application. In this case, instead of actually launching the applications, launch setting is performed for the applications.

As mentioned above, according to the compound

machine 100 of the third embodiment, since the applications that are mutually indispensable can be executed at the same time, the compound machine 100 can operate stably.

5 In the same way as the sixth embodiment, a plurality of VASes 941-948 can be launched for each application in which each VAS performs the indispensable application information obtaining process and the application launch control process for a
10 corresponding application.

Fig.73 is a figure showing a configuration of the VASes 941-948 of the compound machine 100, and relationship among the VASes 941-948, each application, control service layer 150 and general OS 121.

15 (Eighth embodiment).

In the following, the eighth embodiment will be described. The whole configuration of the compound machine 100 of the Eighth embodiment is the same as that of the first embodiment.

20 In the eighth embodiment, the compound machine 100 stores a database that can represent incompatible or unrecommended relationship and indispensable or recommended relationship for applications. The VAS 140 performs launch control by
25 referring to the database. In this specification,

incompatible or unrecommended relationship may be called simply "incompatible relationship", and indispensable or recommended relationship may be called simply "indispensable relationship".

- 5 An example of the database file is shown in Fig.74. As shown in Fig.74, the file shows indispensable or incompatible applications for each application. In the field of "incompatible or indispensable application", the number inside the
- 10 parentheses indicates a level of incompatible relationship or indispensable relationship. The larger the absolute value in the minus side is, the higher the level of incompatible relationship is. The larger the absolute value in the plus side is, the higher the
- 15 level of indispensable relationship is. For example, "-1" means that although both of the applications can be executed simultaneously, it is not recommended to execute both of the applications since there is a possibility that the speed of the system may be lowered.
- 20 In addition, for example, "-3" means that the applications can not be executed simultaneously. In addition, for example, "+1" means that, although each of the applications can be executed solely, more functions can be provided if both of the applications
- 25 are executed simultaneously. In addition, for example,

" +3" means that the applications should be executed simultaneously. As to the incompatible relationship, in addition to the above-mentioned level, priority of launch can be included. The VAS 140 performs launch
5 control by referring to the file shown in Fig.74.

In addition to the information shown in Fig.74, the file may include, for each pair of applications, a reason for being indispensable or incompatible relationship, and other information. By
10 including such information, the reasons can be displayed on the operation panel.

The above-mentioned file can be stored in the compound machine 100 by the user. In addition, the compound machine 100 may obtain the file from a server.
15 In the case when the file is stored in the server, if a problem as to indispensability or incompatibility of applications is found, the problem information can be sent to the server by using a proper format, so that the file can be automatically updated. In addition,
20 when an application is developed and the application is indispensable for another application or is incompatible with another application, such information can be sent to the server, so that the file can be automatically updated.

25 The file is not necessarily stored in the

compound machine 100, and the VAS 140 can refer to the file that exists outside of the compound machine 100.

Next, the procedure of launch control by the compound machine 100 by using the file shown in Fig.74 will be described with reference to the flowchart shown in Fig.75. The following procedure is for a case when launch setting is performed.

First, the VAS 140 displays a screen for launch setting on the operation panel in step S3101. An example of the launch setting screen is shown in Fig.76. The screen shown in Fig.76 shows that launch setting has been already performed for "simple copy".

Next, the user selects an application (to be referred to as "selected application" hereinafter) on which the user wants to perform launch setting. The VAS 140 obtains the product ID of the selected application, and searches the file shown in Fig.74 on the basis of the product ID in step S3102.

Then, the VAS 140 obtains product IDs and related information (such as reason for being incompatible or indispensable relationship) corresponding to the key product ID in step S3103. For example, in the example of Fig.74, if the product ID of the selected application is 101, the VAS 140 obtains 102(-3) and 103(+1).

Then, the VAS 140 performs launch control according to the obtained information.

If the application corresponding to the product ID obtained from the file is an incompatible application against the selected application, that is, 5 if the level is minus (Y in step S3104), the VAS 140 checks if launch setting has been performed for the incompatible application in step S3105. If launch setting has not been performed for the incompatible 10 application, launch setting is performed for the selected application in step S3106.

If launch setting has been performed for the incompatible application (Y in step S3105), the level is checked in step S3107. Then, the VAS 140 displays a 15 warning screen (S3108, if the level is low), or display a selection screen for selecting an application on which launch setting is performed between the incompatible application and the selected application (S3109). The warning screen shows information that 20 inquires the user whether launch setting can be allowed for both of the applications. When the warning screen is displayed, launch setting is stopped for one of the selected application and the incompatible application or launch setting is performed for both of the 25 applications in step S3110. When the selection screen

is displayed, launch setting is performed on an application selected in the selection screen in step S3111. In this case, if the selected application is selected, launch setting for the incompatible
5 application is released.

When an application corresponding to the obtained product ID is indispensable for the selected application (N in step S3104), the VAS 140 checks whether launch setting has been performed on the
10 indispensable application in step S3112. If launch setting has been performed on the indispensable application, launch setting of the selected application is performed in step S3113. If launch setting has not been performed on the indispensable application, next
15 process is performed (N in step S3112).

The level of indispensable relationship is checked in step S3114. If the level is high so that the both applications should be executed simultaneously, the VAS 140 displays a screen for prompting the user to
20 perform launch setting on the indispensable application in step S3115. Then, launch setting is performed on the indispensable application according to determination of the user in step S3116. In this case, launch setting is performed for both of the selected
25 application and the indispensable application.

For example, in the screen shown in Fig.76, assuming that "simple scanner" is the indispensable application for the "edit print tool", when "edit print tool" is selected for performing launch setting as shown in Fig.77, a screen shown in Fig.78 is shown since launch setting has not been performed on "simple scanner". When the user pushes the OK button, launch setting is performed on both of "edit print tool" and "simple scanner".

10 If the level of indispensable relationship is low in step S3114, that is, if the level indicates recommended relationship (for example, extension can be provided by another application), the VAS 140 displays a screen for notifying the user that the recommended application can add extension to the selected application in step S3117. Then, the VAS 140 performs launch setting on both of the selected application and the corresponding recommended application in step S3118.

20 For example, in the example of Fig.77, if "user stamp add module" and "user comment add module" can add extension to "edit print tool", the screen shown in Fig.79 is displayed. If OK button is pushed, launch setting is performed for the three applications.

For performing launch setting for applications of indispensable relationship, launch

25

setting can be performed for the applications without inquiring the user.

When the indispensable application that should be launched with the selected application is not
5 installed in the compound machine 100, a screen of Fig.80 is displayed before displaying the screen of Fig.78, so that the indispensable application can be downloaded. Instead of displaying the screen for download, a screen showing reference of the application
10 can be displayed as shown in Fig.81.

In the above-mentioned example, in a case when an application that is selected for performing launch setting is incompatible with another application on which launch setting has been performed, the VAS 140
15 can refer to the priorities between the selected application and the another application, and can automatically perform launch setting for an application that has a higher priority. For example, if the priority of the selected application is higher than the
20 another application, the VAS 140 can release the launch setting for the another application and perform launch setting for the selected application.

Although the above-mentioned example is for a case when launch setting is performed, the above-
25 mentioned method can be applied to a case when an

application is actually launched.

As mentioned above, according to the present invention, an image forming apparatus in which applications can be installed is provided, and the
5 image forming apparatus includes: an application information obtaining part for obtaining application information that is used for determining whether an application is installable in the image forming apparatus.

10 According to the present invention, it can be determined whether an application is installable.

In the image forming apparatus, the application information obtaining part obtains the application information that is included in an
15 execution file of the application by using interprocess communication between the application information obtaining part and the application.

The application information obtaining part may obtain the application information from a file
20 separate from the application. For example, the application information can be obtained via a network.

In the image forming apparatus, the application causes the image forming apparatus to perform: a step of determining whether to provisionally
25 launch the application or to normally launch the

application according to a request from the application
information obtaining part; a step of providing the
application information to the application information
obtaining part if the application is provisionally
5 launched. There, normal launch or provisional launch
can be selected.

The image forming apparatus may further
includes a storage for storing the application
information obtained by the application information
10 obtaining part. The storage is at least one of a hard
disk, a nonvolatile memory and a memory card, and the
application information obtaining part obtains the
application information by reading information stored
in the storage.

15 In addition, the application information
includes a product ID that is specific for the
application. The application information may further
includes at least one of a vendor name, an application
name, version information and resource information.

20 In the image forming apparatus, the storage
stores product IDs of installable applications, and the
image forming apparatus determines whether a target
application is installable by checking whether a
product ID of the target application is stored. Thus,
25 it becomes easy to determine whether an application is

installable.

In addition, according to the present invention, an image forming apparatus including an operation panel for displaying or inputting operation
5 information is provided, and the image forming apparatus includes:

an information obtaining part for obtaining application information on an application to be used in the image forming apparatus; and

10 a display part for displaying the application information or information related to the application information on the operation panel.

According to the present invention, since application information is displayed on the operation
15 panel, the user can determine whether an application is installable.

The image forming apparatus may further includes a determination part for determining whether the application is installable in the image forming
20 apparatus according to the application information obtained by the information obtaining part.

Accordingly, the image forming apparatus can automatically determine whether an application is installable. The display part can display one or more
25 installable applications on the operation panel

according to the result of determination by the determination part, wherein an installable application can be selected from the one or more installable applications.

5 In addition, the display part can display a reason why an application is not installable according to the result of determination by the determination part. According to this invention, the user can know the reason why the application is not installable. For
10 example, if an application is installable due lack of a hardware resource, the user can know the reason.

 In addition, the display part can display a warning when an application selected for install is already installed in the image forming apparatus.

15 In the image forming apparatus, the application information or the information related to the application information that is displayed on the operation panel can be information necessary for determining whether the application is installable.

20 According to this configuration, the user can determines whether an application is installable.

 The image forming apparatus may further include a part for causing a remote apparatus to display a screen that is the same as a screen displayed
25 on the operation panel. Accordingly, the user can

operate the image forming apparatus from the remote apparatus.

In addition, according to the present invention, an image forming apparatus in which a plurality of applications can be installed is provided, in which the image forming apparatus includes:

an information obtaining part for obtaining necessary resource information for executing an application and available resource information in the image forming apparatus; and

a launch restriction part for comparing the necessary resource information and the available resource information, and performing launch restriction process on the application according to the result of comparison between the necessary resource information and the available resource information.

According to the present invention, the image forming apparatus can prevent erroneous launch of an application for which resources are lacking. Thus, the image forming can operate stably.

In the image forming apparatus, the image obtaining part can obtain, as the necessary resource information, a usage record of a resource used by the application. Therefore, accurate information can be obtained. The information obtaining part can obtain

the usage record from system information that holds usage records used by processes in the image forming apparatus.

In the image forming apparatus, the necessary
5 resource information indicates a necessary resource amount and the available resource information indicates an available resource amount, and wherein the launch restriction part can display a ratio of the necessary resource amount to the available resource amount on an
10 operation panel. According to this configuration, the user can grasp the ratio of the necessary resource amount to the available resource amount.

In addition, the launch restriction part can display a warning on an operation panel when the
15 necessary resource amount is larger than the available resource amount. Therefore, the user can know lack of resources for an application.

In the image forming apparatus, if the application can operate by using a second resource that
20 is different from a first resource corresponding to the necessary resource amount, the launch restriction part causes the application to use the second resource instead of the first resource. Therefore, even when a resource is lacking, the application can be executed by
25 using another resource.

In addition, according to the present invention, an image forming apparatus in which a plurality of applications can be installed is provided in which the image forming apparatus includes:

5 a part for referring to a database which includes, for each application, information on propriety of combination of applications; and

 a launch control part for performing launch control process on an application according to the
10 information on propriety.

 According to the present invention, proper applications can be launched according to the information on propriety of combination of applications.

 In the image forming apparatus, the
15 information on propriety indicates incompatible relationship or unrecommended relationship among a plurality of applications, and the launch control part can display a screen for selecting at least one application from among the plurality of applications on
20 an operation panel. Therefore, the user can select a desired application among mutually incompatible applications.

 The launch control part may launch at least one application among the plurality of applications or
25 performs launch setting on at least one application

among the plurality of applications. According to this configuration, the image forming apparatus can prevent mutually incompatible applications from being launched simultaneously.

5 In addition, the launch control part may display a warning screen on an operation panel when combination of applications is unrecommended. Since the warning screen is displayed, the user can determines whether to launch one application or both of
10 the applications.

 In the image forming apparatus, the information on propriety can indicate a level of the incompatible relationship or the unrecommended relationship, in which the launch control part can
15 perform the launch control process according to the level.

 In the image forming apparatus, the information on propriety may indicate indispensable relationship or recommended relationship among a
20 plurality of applications, wherein the launch control part can display a screen showing an indispensable or recommended application for another application on an operation panel. According to this configuration, the user can select the indispensable or recommended
25 application.

In addition, launch control part can launch an indispensable or recommended application for another application, or performs launch setting on the indispensable or recommended application for another application. When the indispensable or recommended application is not installed in the image forming apparatus, the launch control part can download the indispensable or recommended application from a server via a network. Also, the launch control part can display information that can be used for obtaining the indispensable or recommended application.

In the image forming apparatus, the information on propriety can further indicate a level of the indispensable relationship or the recommended relationship, in which the launch control part can perform the launch control process according to the level.

The present invention is not limited to the specifically disclosed embodiments, and variations and modifications may be made without departing from the scope of the present invention.